

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

S00P10874500
Priority Paper
2/13/01
MOA
JC658 U.S. PTO
09/661878
09/14/00


別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日

Date of Application: 1999年 9月14日

出願番号

Application Number: 平成11年特許願第261262号

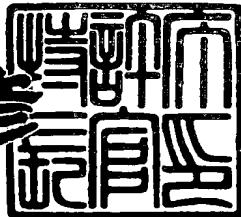
出願人

Applicant(s): ソニー株式会社

2000年 6月29日

特許庁長官
Commissioner,
Patent Office

近藤 隆彦



出証番号 出証特2000-3050041

【書類名】 特許願
【整理番号】 9900066903
【提出日】 平成11年 9月14日
【あて先】 特許庁長官殿
【国際特許分類】 H04N 7/133
【発明者】
【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社
内
【氏名】 カーティス ユーバンクス
【特許出願人】
【識別番号】 000002185
【氏名又は名称】 ソニー株式会社
【代表者】 出井 伸之
【代理人】
【識別番号】 100067736
【弁理士】
【氏名又は名称】 小池 晃
【選任した代理人】
【識別番号】 100086335
【弁理士】
【氏名又は名称】 田村 榮一
【選任した代理人】
【識別番号】 100096677
【弁理士】
【氏名又は名称】 伊賀 誠司
【手数料の表示】
【予納台帳番号】 019530
【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9707387

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 信号処理装置及び方法並びに記録媒体

【特許請求の範囲】

- 【請求項1】 入力画像信号の形式を特定する形式特定手段と、
上記入力画像信号に対する編集操作が入力される入力手段と、
上記入力画像信号を上記入力手段に入力された編集操作にしたがって編集処理
する、上記形式に対応する複数の編集処理手段と、
上記編集処理手段にて編集された上記入力画像信号を出力ストリームに変換す
る変換手段と、
上記形式特定手段にて特定した上記入力画像信号の形式に応じて、当該形式に
対応する上記編集処理手段で編集処理を行うように制御する制御手段と
を有することを特徴とする信号処理装置。
- 【請求項2】 上記形式とは、上記入力画像信号のフォーマット又はフォーマ
ットのクラスであることを特徴とする請求項1記載の信号処理装置。
- 【請求項3】 上記編集操作は、上記入力画像信号のカット、コピー、ペース
ト、リサイズ、時間的な移動、テキスト又は画像の重畠、トランジションを含む
ことを特徴とする請求項1記載の信号処理装置。
- 【請求項4】 上記編集処理手段における、特定の形式に対する編集処理を置
換、削除又は置換する編集処理管理手段を有することを特徴とする請求項1記載
の信号処理装置。
- 【請求項5】 上記入力画像信号を共通フォーマットに変換するフォーマット
変換手段を有し、上記編集処理手段は、上記フォーマット変換手段で共通フォー
マットに変換された画像信号に対して編集処理を施すことを特徴とする請求項1
記載の信号処理装置。

- 【請求項6】 入力画像信号の形式を特定し、
上記入力画像信号に対する編集操作が入力され、
上記入力画像信号を入力された編集操作にしたがって編集し、
編集された上記入力画像信号を出力ストリームに変換し、
特定した上記入力画像信号の形式に応じて、当該形式に対応する編集を行うよ

うに制御すること

を特徴とする信号処理方法。

【請求項7】 上記形式とは、上記入力画像信号のフォーマット又はフォーマットのクラスであることを特徴とする請求項6記載の信号処理方法。

【請求項8】 上記入力画像信号を共通フォーマットに変換することを特徴とする請求項6記載の信号処理方法。

【請求項9】 画像信号を処理するプログラムが記録された記録媒体であって、上記プログラムは、

入力画像信号の形式を特定し、

上記入力画像信号に対する編集操作が入力され、

上記入力画像信号を入力された編集操作にしたがって編集し、

編集された上記入力画像信号を出力ストリームに変換し、

特定した上記入力画像信号の形式に応じて、当該形式に対応する編集を行うように制御すること

を特徴とする記録媒体。

【請求項10】 上記形式とは、上記入力画像信号のフォーマット又はフォーマットのクラスであることを特徴とする請求項9記載の記録媒体。

【請求項11】 上記入力画像信号を共通フォーマットに変換することを特徴とする請求項9記載の記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、画像信号を処理する信号処理装置及び方法並びに画像信号を処理するプログラムが記録された記録媒体に関する。

【0002】

【従来の技術】

従来、非線形ビデオ編集装置と称される画像編集装置が提供されている。この非線形ビデオ編集装置は、編集の際の一時記録媒体としてハードディスクの如きディスク状記録媒体を備えるので、ランダムアクセスが可能である。したがって

、シーケンシャルではない非線形な編集が可能である。かかる非線形ビデオ編集装置においては、所望の映像素材の迅速な読み出しと、一連の映像素材に対する素材の挿入及び削除を容易に行うことができる。

【0003】

【発明が解決しようとする課題】

しかし、従来の非線形ビデオ編集装置は、ユーザーに単一の画像データのフォーマットを使用するように制限するか、又は複数のフォーマットの使用を可能とさせるが、特定の効果又は全部の効果（例えば、トランジション効果）の画像処理を行う場合に、画像データを共通フォーマットに変換するように制限していた。

【0004】

前者の場合には、単一フォーマットを使用するように制限すると、ユーザがフォーマットを自由に選択する自由を妨げることになり、ユーザへ負担を掛けることになる。

【0005】

また、後者の場合には、共通フォーマットに変換する処理は、顕著な処理上のオーバーヘッドが導入されたり、高価なハードウェアが必要とされたり、ビデオ画像の品質の劣化が生じたりする。

【0006】

特に、いわゆるMPEG (Moving Picture Experts Group)、いわゆるJPEG (Joint Photographic Coding Experts Group)、いわゆるH.261、いわゆるDV (Digital Video) 及びその他の通常用いられるデジタル画像フォーマットのように、離散コサイン変換 (Discrete Cosine Transformation; DCT) に基づく画像フォーマットは、圧縮処理の一部としてDCT係数の量子化に依存している。

【0007】

汎用エディタの普通の技術では、このような画像フォーマットを画素 (picture element; pixel) 表示に復元し、空間領域内で画像処理を実行し、その画像を再び圧縮符号化する。しかし、符号化の際におけるDCT係数の再量子化と整数

化は、画像歪みをもたらし、画像歪みは世代を重ねるごとに累積してゆく。

【0008】

本発明は、上述の実情に鑑みて提案されるものであって、画像信号の編集を高速に、画質を劣化させないよう行える信号処理装置及び方法並びにかかる編集のプログラムが記録された記録媒体に関する。

【0009】

【課題を解決するための手段】

上述の課題を解決するために、本発明に係る信号処理装置は、入力画像信号の形式を特定する形式特定手段と、上記入力画像信号に対する編集操作を入力される入力手段と、上記入力信号を上記入力手段に入力された編集操作にしたがって編集処理する、上記形式に対応する複数の編集処理手段と、上記編集処理手段にて編集された上記入力画像信号を出力ストリームに変換する変換手段と、上記形式特定手段にて特定した上記入力信号の形式に応じて、当該形式に対応する上記編集処理手段で編集処理を行うように制御する制御手段とを有するものである。

【0010】

本発明に係る信号処理装置は、入力画像信号の形式を特定し、上記入力画像信号に対する編集操作を入力され、上記入力画像信号を入力された編集操作にしたがって編集し、編集された上記入力画像信号を出力ストリームに変換し、特定した上記入力画像信号の形式に応じて、当該形式に対応する編集を行うように制御するものである。

【0011】

本発明に係る記録媒体は、画像信号を処理するプログラムが記録された記録媒体であって、上記プログラムは、入力画像信号の形式を特定し、上記入力画像信号に対する編集操作を入力され、上記入力画像信号を入力された編集操作にしたがって編集し、編集された上記入力画像信号を出力ストリームに変換し、特定した上記入力画像信号の形式に応じて、当該形式に対応する編集を行うように制御するものである。

【0012】

【発明の実施の形態】

以下、本発明の実施の形態について、図面を参照して説明する。まず、本発明の実施の形態としての画像編集装置の構成を、図1を参照して説明する。本実施の形態の画像編集装置は、複数の画像データのフォーマットを取り扱うことができるものであり、フォーマットに応じて拡張が可能である。

【0013】

この画像編集装置は、特定のフォーマット又はフォーマットのクラス、すなわち、MPEGのようなDCTに基づいた周波数領域及び圧縮領域での処理のための高度に最適化された処理に対応する複数のビデオフォーマットモジュールを備える。この画像編集装置においては、圧縮領域と周波数領域における処理技術を用いることによって、多くの画像処理をより速く、少ない劣化にて行うことができる。

【0014】

図1に示すように、本実施の形態の画像編集装置の好適なハードウェア構成は、画像編集装置の全体を制御するCPU1と、高速のメモリであるキャッシュ2と、システムに対する制御を行うシステム制御部3と、データを記憶するメモリ4とを有している。

【0015】

CPU1は、この画像編集装置の全体を制御すると共に、各種の処理を実行する。キャッシュ2は、高速メモリであって、頻繁にアクセスされるメモリ4の部分を含み、これによってメモリ4へのアクセス速度を増加させる。システム制御部3は、CPU1、キャッシュ2、メモリ4を含むシステムの資源を制御するために使用される。

【0016】

メモリ4にはデータが格納される。メモリ4は、CPU1及びシステム制御部3からのコマンドに基づいて、データを書き込まれたり読み出されたりすることができます。本実施の形態においては、メモリ4にはDRAMが使用される。

【0017】

CPU1、キャッシュ2、システム制御部3及びメモリ4は、内部バス11にて相互に接続されている。内部バス11は、これらの間の高速通信手段として作用する。

【0018】

また、画像編集装置は、大容量の記録装置であるハードディスク6と、ハードディスク6を制御するディスク制御部5と、ポインティングデバイスであるマウス8と、マウス8を制御するマウス制御部7と、文字や画像を表示するモニタ10と、モニタ10の表示を制御する画像制御部9とを有している。

【0019】

ハードディスク6は、固定ディスクを用いた大容量の記録装置である。ディスク制御部5は、ハードディスク6の指定された領域に対するデータの書き込みと読み出しを制御する。

【0020】

画像編集装置においては、ハードディスク6を用いて画像データにランダムアクセスすることが可能であり、画像データに対するシーケンシャルでない非線形編集を行うことができる。

【0021】

マウス8は、その動きに基づいた位置データが得られるポインティングデバイスである。また、マウス8は、ボタンを押圧されることによってもデータが入力される。CPU1は、マウス8から送られたデータに基づいて、モニタ10上にカーソルを表示して、ユーザーとの間で双方向的に制御を行う。

【0022】

モニタ10は、画像を表示するものであり、例えばCRTが利用される。モニタ10は、CPU1によって作成された状態に基づいて、時間変化する画像を表示する。画像制御部9は、CPU1から描画コマンド又は他のコマンドを受け取って処理し、画像信号に変換してモニタ10に送る。

【0023】

システム制御部3、ディスク制御部5、マウス制御部7及び画像制御部9は、

外部バス12にて相互に接続されている。外部バス12は、内部バス11とは別の通信手段として作用する。C P U 1は、システム制御部3を介して、外部バス12上の資源にアクセスする。

【0024】

本実施の形態は、拡張可能な非線形画像編集装置である。この画像編集装置は、入力映像データとして、画像編集装置に既知の任意のフォーマットを受け取り、種々の編集コマンドに応じて映像の出力ストリームを生成することができる。

【0025】

この画像編集装置は、動的に拡張可能なモジュールの組を有している。これらのモジュールは、特定のデータフォーマット又はデータフォーマットのクラスに対して、速度及び／又は画像品質を最適化するように処理することができる。かかるモジュールは、特定のフォーマット又はフォーマットのクラスに特有の特徴に基づく技術を用いている。この技術は、周波数領域と圧縮領域における処理技術を含むが、これに限定されない。

【0026】

フォーマットモジュールは、標準的なビデオ編集A P Iに準拠している。モジュールは、システムに対して、追加され、削除され、又は置き換えられてもよい。

【0027】

画像編集装置は、ビデオモジュールがデータに最適な処理を行うことかできない場合には、ビデオデータを共通フォーマットに変換する。

【0028】

モジュールは、ビデオデータをデータフォーマットのあるクラス中で処理する場合に、そのデータを中間フォームに変換してもよい。

【0029】

映像データの本来のフォーマット中でビデオ処理操作を実行することは、今後、ウェーブレットやフラクタルに基づくフォーマットのような他のビデオフォーマットに対して有効であることがわかるであろう。

【0030】

この画像編集装置は、拡張可能であるので、このような新しい展開が起きた場合には、対応するビデオモジュールをシステムに追加することによって、それらのフォーマットを利用することができる。

【0031】

画像編集装置においては、ハードディスク6に蓄積された画像データに対して非線形の画像処理が行われる。この非線形編集について、図2に示す状態遷移図を参照して説明する。

【0032】

画像データの非線形編集は、画像編集装置のハードウェアで実行されるオペレーティングシステム上で実現される。このオペレーティングシステムは、ファイルシステムとファイル操作に対するシステムレベルの操作、ファイル選択を含むグラフィカルユーザーインターフェース操作、ウィンドウ及びイベント処理、画像処理ファイルに対する標準操作を提供するメディアライブラリを提供する。

【0033】

メディアライブラリは、画像ファイルを、開き、読み出し、書き込み、閉じ、同定することができるものでなければならない。

【0034】

図2において、状態S01では、画像編集装置において、編集コマンドが、グラフィカルユーザーインターフェースにより、又はバッチ処理により入力される。入力された編集コマンドを編集リストに記録する。

【0035】

通常の編集コマンドとしては、入力ファイルをインポートし、インポートした画像ファイルに対する画像処理等がある。具体的には、編集コマンドとしては、カット内の始点と終点を指定し、タイトルや他の画像を重複し、カットとカットとの間におけるワイプとディソルブのようなトランジション（遷移）を指定するためのものがある。

【0036】

画像データに対する標準操作を提供するメディアファイルは、オペレーティン

グシステムによって提供されるファイルダイアログを用いて特定される。画像データ等のマルチメディアデータのコピー、カット、削除、移動、クリッピング等の編集操作は、ユーザーによってグラフィカルインターフェース上でマウスとメニュー命令を用いて指定される。

【0037】

「出力生成」コマンドが与えられると、編集リストが生成される。また、その時にリストを作成してもよいが、ずっとコマンドを受けている時 (event loop: イベント ループ) に動的リスト作ってもよい。

【0038】

編集リストは、いくつかのトラックから構成され、各トラックは、画像、音声、効果のようなトラックタイプとクリップのリストを有している。クリップは、画像データ、音声データ又は効果データと、始点と期間の情報を含んでいる。

【0039】

ここで、編集リストの具体例として、2個の画像トラックと1個の効果トラックに対応するものについて、図3を参照して説明する。この編集リストは、C++を用いたクラス (class) として記述されたものである。しかし、編集リストとこれに伴う構成は、種々の他の同等な方法により表現することが可能である。

【0040】

図3の3Aに示すように、クラス (class) 編集リスト (EditList) は、絶対時間での開始フレームであるstartTimeと、フレーム単位での全ビデオの期間であるdurationと、Aトラック中のクリップの数であるaClipCountと、Aトラックの各クリップのクリップデータであるaSourcesと、Bトラック中のクリップの数であるbClipCountと、Bトラックの各クリップのクリップデータであるbSourcesと、効果トラックの効果の数であるeffectsCountと、効果データであるeffectsとを有している。

【0041】

図3の3Bに示すように、クラス (class) AVクリップ (AVclip) は、絶対フレームでの開始時間であるstartTimeと、このクリップのフレームの数であるdurationと、ビデオ圧縮フォーマットを表すmediaTypeと、この入力ストリームの

ファイル名であるfilenameと、ファイルの開始フレームであるframeOffsetとを有している。

【0042】

図3の3Cにおいては、クラス(class)効果(Effect)は、絶対フレームでの開始時刻であるstartTimeと、この効果のフレームの数であるdurationと、標準フォーマットの効果のタイプ(ワイプ等)を表すeffectTypeと、種々の効果パラメータ(スタート百分比、エンド百分比、方向等)であるeffectParamsとを有している。

【0043】

編集リストは、複数のトラックの間の関係を示すものであるので、時間軸に沿ってグラフィック表示することができる。図4は、編集リストとして、上述のように2個の画像トラックと1個の効果トラックに対応する具体例を示している。

【0044】

図4においては、図4の4Aには第1の画像トラックが、図4の4Bには効果トラックが、図4の4Cには第2の画像トラックが、図4の4Dには時間軸が表示されている。

【0045】

図4の4Aの第1の画像トラックには、時刻t0から時刻t3までの期間にわたって、第1のビデオクリップaが配置されている。

【0046】

図4の4Bの効果トラックにおいては、時刻t0から時刻t1までの期間にわたってタイトルマスクb1が、時刻t2からt3までの期間にわたってワイプトランジションb2が配置されている。

【0047】

図4の4Cの第2の画像トラックには、時刻t2から時刻t4までの期間にわたって第2のビデオクリップcが配置されている。

【0048】

図4の4Dには、時間の経過の方向に時刻t0, t1, t2, t3, t4の順序で表示した時間軸が表示されている。

【0049】

このグラフィック表示においては、第1の画像トラックに含まれるビデオクリップaと、第2の画像トラックに含まれるビデオクリップcとは、画像出力を生成するために結合されている。すなわち、第1の画像トラックに含まれるビデオクリップaの終わりが第2の画像トラックに含まれるビデオクリップcの始まりとオーバーラップする部分に、ワイプトランジションb2が挿入されている。

【0050】

また、ビデオクリップaの始まりには、タイトルマスクb1が重畠された画像出力が得られる。

【0051】

状態S02においては、状態S01において編集リストを編集セグメントに分解する処理が行われる。

【0052】

上述したように、記録した編集リストは、時間軸方向に複数のトラックから構成される。この編集リストにおいて、編集トラック全体を時間軸に沿って分割した部分を編集セグメントという。したがって、編集リストは、時間方向に編集セグメントに分割することができる。

【0053】

編集セグメントは、編集リストを構成する各映像トラックと各効果トラックが单一の値を取る最長の部分として定義される。したがって、図4のグラフィック表示に示した編集リストは、四つの編集セグメントに分割される。

【0054】

分割された編集セグメントは、時刻t0から時刻t1までの期間の第1の編集セグメント、時刻t1から時刻t2までの期間の第2の編集セグメント、時刻t2から時刻t3までの期間の第3の編集セグメント、時刻t3から時刻t4までの期間の第4の編集セグメントになる。

【0055】

ここで、任意の又は全てのトラックデータと効果トラックデータは、編集セグメント単位で空(φ)となり得ることに注意する。

【0056】

このように、図4に示した編集リストのグラフィック表示の例を編集セグメントに分割すると、図5に示すようになる。

【0057】

すなわち、図5の5Aに示す第1の編集セグメント (edit segment 1) は、時刻 t_0 から時刻 t_1 までに対応するもので、開始 (Start) t_0 、期間 (Duration) $t_1 - t_0$ 、第1のビデオ (Video 1) a、第2のビデオ (Video 2) ϕ 、効果 (Effect) b1 である。ここで、 ϕ は対象が存在しない「空」であることを示している。

【0058】

図5の5Bに示す第2の編集セグメント (edit segment 2) は、時刻 t_1 から時刻 t_2 までに対応するもので、開始 t_1 、期間 $t_2 - t_1$ 、第1のビデオ a、第2のビデオ ϕ 、効果 ϕ である。

【0059】

図5の5Cに示す第3の編集セグメント (edit segment 3) は、時刻 t_2 から時刻 t_3 までに対応するもので、開始 t_2 、期間 $t_3 - t_2$ 、第1のビデオ a、第2のビデオ c、効果 b2 である。

【0060】

図5の5Dに示す第4の編集セグメント (edit segment 4) は、時刻 t_3 から時刻 t_4 までに対応するもので、開始 t_3 、期間 $t_4 - t_3$ 、第1のビデオ ϕ 、第2のビデオ c、効果 ϕ である。

【0061】

編集セグメントが一旦構築されると、映像出力の生成を開始することができる。

【0062】

本実施の形態は、状態 S03 で行う処理において、映像出力の生成を、フォーマットや、フォーマットのクラスに対応させて行うものである。このような方法を、図6を参照して説明する。

【0063】

図6の右側のS14は、対応したフォーマット内で映像出力を本実施の形態の編集方法である。図6の左側は、例えば3原色信号からなるRGB信号や、輝度信号と二つの色差信号からなるYCrCb信号のようなベースバンド信号に復号してから映像出力を生成する従来の方法である。

【0064】

ステップS14においては、入力フォーマットが出力フォーマットと同一であるか判断する。入力フォーマットが出力フォーマットと同一であると判断されるとステップS16に進み、同一ではないと判断されると従来の方法によるステップS10に進む。

【0065】

ステップS16においては、出力フォーマットが入力フォーマットと同じであるので、画像編集装置はそのフォーマットに対応するビデオモジュール(VideoModule)を探し、そのビデオモジュールに効果処理を引き渡す。

【0066】

対応するビデオモジュールの探索は、通常は容易な操作である。すなわち、全てのビデオモジュールのリストを探して、ビデオモジュールのフォーマットコードとビデオファイルに埋め込まれたオペレーティングシステムが提供する標準ビデオフォーマット(又は圧縮フォーマット)との比較を繰り返すことによって実現される。

【0067】

ステップS17においては、そのビデオモジュールにおける、効果を適用する関数であるApplyEffectを呼び出す。

【0068】

ここで、ビデオモジュールは、画像処理用の標準API(application program interface)を有する動的にロードされるライブラリである。図7はクラス(class)ビデオモジュール(VideoModule)のインターフェースである。

【0069】

このビデオモジュールは、標準フォーマット特定子であるフォーマット(format)

at) と、標準クラス特定子であるクラス (class) と、フレームをRGBに復号するConvertToBasebandと、フレームをフォーマットに復号する（例えばDCT レベルに）ConvertToClassと、このフォーマットの知識を用いてトランジション効果を適用するApplyEffectとを有している。

【0070】

本実施の形態においては、フォーマットは、例えば、いわゆるDV又はいわゆるMPEG-2のような公開された標準フォーマットを表す整数である。クラスは、ビデオモジュールが属するフォーマットのグループを示す。例えばDVとmotion JPEGは、共に「I-フレームのみのDCTに基づくビデオフォーマット」のクラスに属する。あるクラス内のビデオフォーマットはある中間フォームに変換することができる。この中間フォームは、速度、スペース及び／又は品質の点に関して圧縮されていないフォーマットよりも好ましい。したがって、DVとmotion JPEGは、共に量子化されていないDCT係数から構成されるフォーマットに変換することができる。

【0071】

なお、従来では、効果を持たない編集セグメントに対しては、画像編集装置は、フレームに基づいたビデオフォーマットの全フレームの圧縮領域におけるコピーを行う。一方、効果を持つ編集セグメントに対しては、セグメント内の各フレームを画素領域に複合して画素領域で効果を適用され、その後にそのフレームが再圧縮されて出力ストリームに加えられる。ビデオ出力の全部が完成するまで、この処理が各編集セグメントについて続行される。

【0072】

一方、従来の方法による出力映像の生成は、ステップS10において行われる。

【0073】

ステップS10において、ステップS11において入力フレームを通常のRGBフォーマットに復号し、ステップS12においてRGB領域で効果を実行し、ステップS13においてフレームを出力フォーマットに再符号化する。

【0074】

編集装置における各フレームのベースバンドへの復号は、符号化／復号器（coder/decoder; codec）を用いることによって行われる。

【0075】

上述したような、フォーマットに対応して映像出力を生成する方法は、さらにフォーマットのクラスに対応して映像出力を生成するように拡張することができる。フォーマットのクラスを用いる一つの編集セグメントの処理は、図8に示すようになる。

【0076】

ステップS20においては、入力フォーマットを所望の出力フォーマットと比較する。これらが同一であればステップS21に進み、同一でなければステップS23に進む。

【0077】

入力フォーマットが出力フォーマットと同一であれば、図6におけるS16とS17で記載した同じ方法により処理を継続する。すなわち、ステップS21においては、ステップS16と同様に、対応するビデオモジュールを探索する。ステップS22においては、ステップS17と同様に、ビデオモジュールのApplyEffectを用いて効果を適用する。

【0078】

一方、ステップS23においては、入力クラスは出力クラスと比較される。クラスも異なるなら、図6に示したステップS10で前に示したように、入力フレームをベースバンド信号を用いて画素領域で結合する従来の方法に進む。

【0079】

クラスが同一であれば、ステップS24に進み、入力を中間フォームに変換する。

【0080】

上述したように、フォーマットのクラスは、ビデオモジュールが属するフォーマットのグループを示すものであり、例えばDVとmotion JPEGは、共に「I-フレームのみのDCTに基づくビデオフォーマット」のクラスに属する。

【0081】

これらのフォーマットのクラスが共通する画像データは、共通フォーマットである中間フォームに変換される。

【0082】

ステップS25において、この中間フォーム上で、ビデオモジュール (VideoModule) のApplyEffectによって処理が適用される。最後に、ステップS25の結果は、ステップS26において中間フォームから出力フォーマットに変換される。

【0083】

続いて、図9、図10、図11を参照して、画像編集の処理のより詳細な例について説明する。

【0084】

図9のように、異なったフォーマットの二つの入力ファイルがディソルブ (dissolve) トランジションを用いて結合され、いわゆるDVファイルとして出力されるものとする。

【0085】

第1の入力「ウェディング-DV. avi」はいわゆるDVフォーマットである。第2の入力「ピクニック-MJPEG. avi」はいわゆるmotion JPEGフォーマットである。そして、これらを結合したいわゆるDVフォーマットにより、「ハッピー・デー-DV. avi」を出力している。

【0086】

図10は、いわゆるDVといわゆるMJPEGフォーマット、それらの共通のフォーマットのクラスであるIフレームのみのDCTフォーマットのクラスを識別するための、いくつかの便利な定数を簡単に列記している。

【0087】

これらの定数は、FourCC(a,b,c,d)によって、24ビットシフトされた第1の数a、16ビットシフトされた第2の数b、8ビットシフトされた第3の数c、及び第4の数dの論理和として定義されてている。

【0088】

すなわち、FORMAT_DVは、FourCC('D','V','C','S')と、FORMAT_MJPGは、FourCC('M','J','P','G')と、FORMAT_IDCTは、FourCC('I','D','C','T')と、それぞれ定義される。

【0089】

このようなフォーマットのクラスを用いた、図9に示したような入力ファイルの結合について、図11を参照して説明する。

【0090】

ステップS30においては、異なるフォーマットであるが同じクラス（I-フレームDCT）の二つのファイル「ウェディング-DV.avi」と「ピクニック-MPEG.avi」が与えられる。これらのファイルは、以下のステップにより、ディソルブ(dissolve)トランジションにて結合される。

【0091】

ステップS31において、二つの入力ファイルのフォーマットを決定する。フォーマットはオペレーティングシステムのメディアライブラリを用いて決定される。これにより、「ウェディング-DV.avi」と「ピクニック-MPEG.avi」のフォーマットは、それぞれ「FORMAT_DV」と「FORMAT_MJPEG」と決定される。

【0092】

これらのフォーマットは同一ではないので、ステップS32に進み、それぞれのフォーマットに対するビデオモジュールを探索する。ビデオモジュールの探索の結果、「ウェディング-DV.avi」と「ピクニック-MJPEG.avi」のビデオモジュールは、それぞれ「DVVideoModule」と「MJPEGVideoModule」となる。

【0093】

ステップS33において、フォーマットが等しくないので、ビデオモジュールにそのクラスを照会し、これらを相互に比較する。すなわち、「ウェディング-DV.avi」と「ピクニック-MJPEG.avi」にそれぞれ対応するクラスモジュール「DVVideoModule」と「MJPEGVideoModule」からクラスを取得する。

取得したクラスは、同じく「CLASS_IDCT」となる。

【0094】

ステップS34においては、ステップS33で求めたクラスが等しいので、入力ビデオモジュール上でConvertToClass()を呼び出すことにより、各入力ビデオを共通クラスフォーマットに変換する。

【0095】

ステップS35において、IDCTVideoModuleが、変換されたDCT形式のデータをディソルブ効果を用いて結合する。

【0096】

ステップS36において、中間フォームは出力フォーマット(DV)に変換される。

【0097】

最後にステップS37において、編集セグメントが目的ファイルに出力される。この処理は従来のビデオ画像編集装置が効果を実行する前にビデオを画素フォーマットに復号する方法と同様である。

【0098】

但し、本発明においては、フレームが部分的に復号されるのみであることが異なる。この部分的復号は、完全なソフトウェア復号よりも通常は迅速に行うことが可能であり、上述のようにフォーマットモジュールのクラスがフォーマット特有の最適化のクラスを利用することを許容する。

【0099】

このように、本実施の形態においては、ビデオモジュールを用いることにより、フォーマットに固有の知識をカプセル化し、速度及び／又は画質について最適化された編集操作を実行する。

【0100】

本実施の形態においては、ビデオモジュールを、ハードウェアディスクの特定ディレクトリに配置された動的にリンクされるライブラリとして実現している。

【0101】

画像編集装置においては、編集装置は、複数のビデオモジュールにアクセスし

て、ビデオモジュールを更新するための手段を有している。ビデオモジュールは、単一の編集セッション、又は編集セッションの間に、動的に更新することができる。すなわち、動的リンクライブラリファイルを追加、削除、置き換えることにより、画像編集装置が利用できるフォーマットは自動的に更新される。

【0102】

次に、DCTに基づくビデオを用いる効果の処理について説明する。ビデオモジュールを用いることにより、本実施の形態は、任意の数の圧縮フォーマットも効果的に支持する。DCTに基づいて行うデジタルビデオ圧縮の利点について述べる。

【0103】

DCTに基づくビデオ圧縮方法は、各フレームを規則的なサイズの（しばしば 8×8 ）画素ブロックに分割する。これらのブロックは画素コンポーネント（例えばY, Cr, Cb）によってマクロブロックに構成される。マクロブロックの2つの例を図12に示す。

【0104】

図12においては、 8×8 のブロックを単位として、4:1:1マクロブロック22と4:2:0マクロブロック23の両方が6つのブロックから構成される。

【0105】

すなわち、4:1:1マクロブロック22は、4個のYブロック22a、1個のCrブロック22b及び1個のCbブロック22cから構成される。また、4:2:0マクロブロック23は、4個のYブロック23a、1個のCrブロック23b及び1個のCbブロック23cから構成される。このように、Yブロックと、CrブロックとCbブロックが重なってマクロブロックになる。

【0106】

図12では2つの例のマクロブロック構造が示されているが、多くの同様なマクロブロック構造が通常使用されている。

【0107】

図13には、ビデオのフレームをマクロブロックに分割する2つの例を示して

いる。

【0108】

すなわち、図13においては、ビデオのフレーム33は、縦横に 14×5 の $4 : 1 : 1$ マクロブロック31から構成されている。 $4 : 1 : 1$ マクロブロック31は、Y信号に対応する4個のYブロック31aと、Cr信号に対応する第5のブロック31bと、Cb信号に対応する第6のブロック31cとから構成されている。

【0109】

また、ビデオのフレーム34は、縦横に 7×10 の $4 : 2 : 0$ マクロブロック32から構成されている。 $4 : 2 : 0$ マクロブロック32は、Y信号に対応する4個のYブロック32aと、Cr信号に対応する第5のマクロブロック32bと、Cb信号に対応する第6のブロック32cとから構成されている。

【0110】

続いて、トランジションの一例としてワイプトランジションについて、図14を参照して説明する。

【0111】

図14に示すワイプトランジションは、文字「B」を表示する第1の入力フレーム41と文字「A」を表示する第2の入力フレーム42とを用いて、出力フレーム43に対して、片方から第1の入力フレーム41と第2の入力フレーム42との一方を拭い去るよう消していく、後から他方のフレームを現していくものである。

【0112】

出力フレーム43においては、第1の入力フレーム41の一部41aと、第2の入力フレーム42の一部42bとが表示されている。

【0113】

出力フレームのトランジションエッジ44に沿ったマクロブロック45は、第1の入力フレーム41と第2の入力フレーム42とから構成されている。これらの2つの入力は、可能なら周波数領域で結合されるが、周波数領域で結合することができないときには必要なブロック数だけ画素領域で結合される。

【0114】

ここで、ワイプ位置を跨がないマクロブロックは、周波数領域でコピーにより処理できる。ワイプ位置を跨ぐ場合にも周波数領域で処理できことが多い。このことについては、後述する。

【0115】

例えば、三次元効果や変形作画 (warping) のような非マクロブロック局所効果を伴うさらに複雑な効果は、周波数領域で実現することが困難なので、従来方法（ベースバンド）で取り扱われる。

【0116】

このワイプトランジションは、多くの場合、入力フレームから圧縮マクロブロックをコピーすることにより実行することができる。トランジションエッジにまたがる少数のマクロブロックは、入力フレームから適当なブロックをDCT係数レベルに部分的に復号し、それらを予め計算されたワイプ行列を用いて結合し、そして最終的なビット・ストリームに再圧縮することで生成される。ある場合には、コピーされたブロックと結合されたブロックは、一つのビットストリームに許容されたビット数以内に収まらないので再量子化が行われる。

【0117】

アイリス効果及びビットマスクオーバーレイのような他の「局所」効果も、大半のマクロブロックをソースから直接にコピーし、残りのマクロブロックについてDCT領域処理技術を用いて、同様の方法で実行することができる。

【0118】

クロスディソルブは入力マクロブロックをコピーすることでは達成できない。なぜなら、それは全てのマクロブロック入力ブロックの線形結合であるからである。しかし、DCTの特有の性質は、クロスディソルブが入力DCT係数の重み付けされた結合を用いて簡単に計算できる。

【0119】

出力DCT係数 $c_{i,j}$ は、次式に示すように計算することができる。

$$c_{i,j} = \alpha \times a_{i,j} + (1 - \alpha) \times b_{i,j}$$

ここで、 $c_{i,j}$ は、量子化されていないDCTの行 i 、列 j の出力係数である。

$a_{i,j}$ は、量子化されていないDCTの行 i 、列 j の入力Aの係数である。 $b_{i,j}$ は、量子化されていないDCTの行 i 、列 j の入力Bの係数である。 α は、入力Aが受けるディソルブの割合であり、 $0 \leq \alpha \leq 1$ 。0の範囲にある。

【0120】

なお、DCT領域での処理技術の優れた概論として、カリフォルニア大学バークレー校の「電気技術及びコンピュータ科学」誌に掲載された、シーフー・チャン (Shih-Fu Chang) 氏の1993年の工学博士号論文「マルチメディア・ネットワーク・ビデオ・サービス用のビデオ信号の合成と操作 (Compositing and Manipulation of Video Signals for Multimedia Network Video Services)」がある。

【0121】

さらに、本発明は、編集装置ハードディスク6に格納されたプログラムとしても実現され、同じくハードディスク6に格納されたメディアファイルを使用する。複数のハードディスク又はその他の記憶用メディアの使用も、本発明の範囲に入る。

【0122】

例えば、図1に示した編集装置におけるハードディスクには、上述した画像編集を行うプログラムが記録される。このプログラムは、メモリ4に読み込まれ、オペレーティングシステム上で実行され、上述したような画像編集装置における編集を実現する。

【0123】

このプログラムは、入力画像信号の形式を特定し、上記入力画像信号に対する編集操作を入力され、上記入力画像信号を入力された編集操作にしたがって編集し、編集された上記入力画像信号を出力ストリームに変換し、特定した上記入力画像信号のフォーマットやフォーマットのクラスに応じて、当該形式に対応する編集を行うように制御するものである。上記入力画像信号を中間フォームに変換しても良い。

【0124】

なお、上述の実施の形態においては、フォーマットとしてDVやmotion JPE

G等を、フォーマットのクラスとしてはI-フレームを例示したが、本発明はこれに限定されない。他のフォーマットやフォーマットのクラスを適用することができるるのはいうまでもない。

【0125】

【発明の効果】

上述のように、本発明によると、種々のフォーマットの圧縮画像信号を伸長することなく編集することができる。したがって、本発明によると、画像信号の編集を高速に、品質を劣化させることなく行うことができる。

【図面の簡単な説明】

【図1】

映像編集装置の全体の構成を概略的に示すブロック図である。

【図2】

映像編集装置における編集処理の手順を示す状態遷移図である。

【図3】

編集リスト、AVクリップ及び効果の構造体を示す図である。

【図4】

簡単なビデオ編集の時系列を示す図である。

【図5】

簡単な編集リストを分割した編集セグメントを示す図である。

【図6】

編集セグメントの処理を示すフローチャートである。

【図7】

クラスビデオモジュールを示す図である。

【図8】

編集セグメントの処理を示すフローチャートである。

【図9】

異なったフォーマットを有する2個の入力ファイルからなる編集セグメントを示す図である。

【図10】

クラスは同じであるがフォーマットが異なる入力ビデオを用いる編集セグメントを処理する例を示す図である。

【図11】

フォーマットとクラスの定義を示す図である。

【図12】

典型的なマクロブロックの構造を示す図である。

【図13】

マクロブロックの画面への配置を示す図である。

【図14】

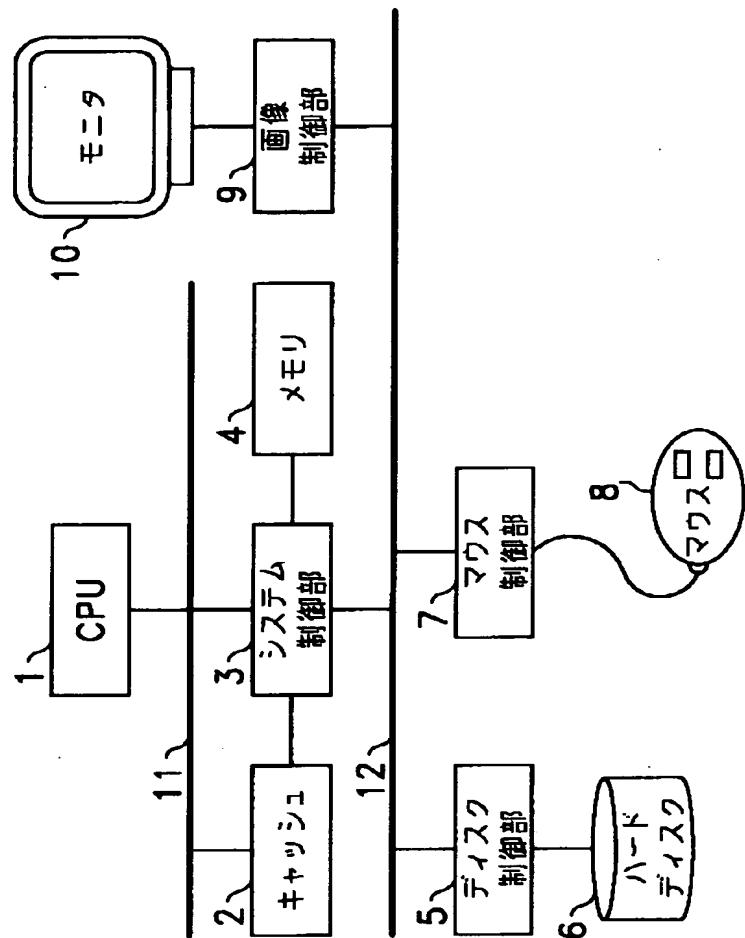
大部分がマクロブロックの複写で構成されるワイプトランジションを示す図である。

【符号の説明】

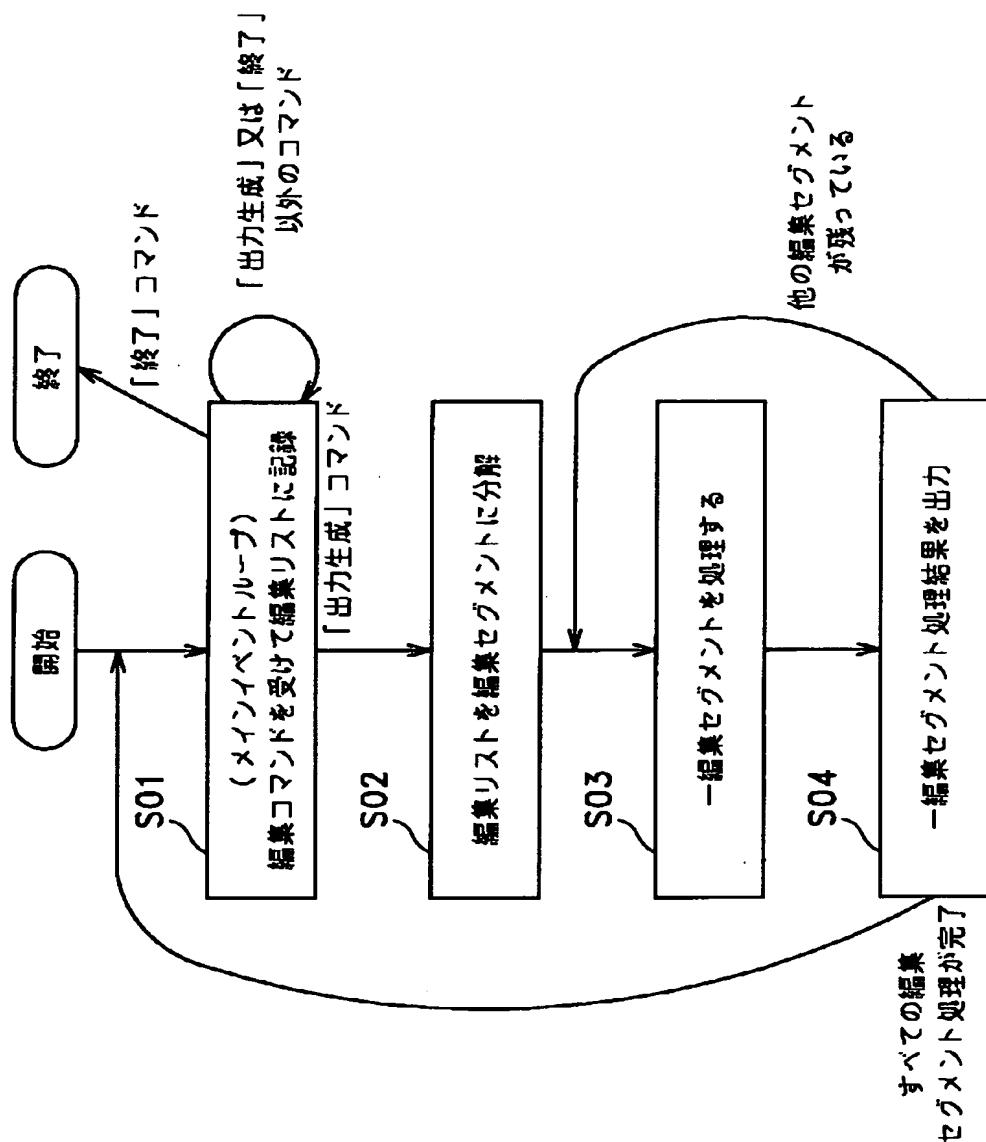
1 CPU、2 キャッシュ、3 システム制御部、4 メモリ、5 ディスク制御部、6 ハードディスク、7 マウス制御部、8 マウス、9 ビデオ制御部、10 モニタ

【書類名】 図面

【図1】



【図2】



【図3】

3A

```
class EditList {
    int startTime;      /* 開始時刻(絶対フレーム) */
    int duration;       /* 全ビデオの期間(フレーム) */
    int aClipCount;    /* Aトラック中のクリップの数 */
    AVClip aSources[]; /* Aトラックの各クリップのクリップデータ */
    int bClipCount;    /* Bトラック中のクリップの数 */
    AVClip bSources[]; /* Bトラックの各クリップのクリップデータ */
    int effectsCount; /* 効果トラックの効果の数 */
    Effect effects[]; /* 効果データ */
};
```

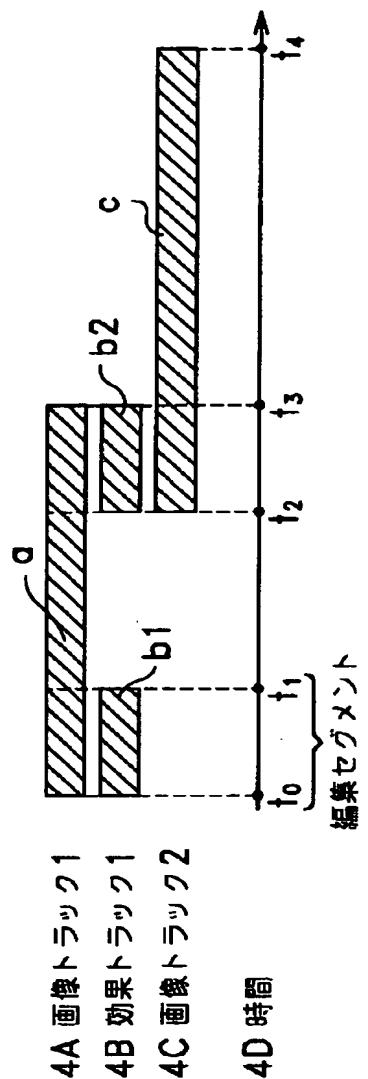
3B

```
class AVClip {
    int startTime;      /* 開始時刻(絶対フレーム) */
    int duration;       /* このクリップのフレームの数 */
    int mediaType;     /* ビデオ圧縮フォーマット(標準) */
    char filename[];   /* この入力ストリームのファイル名 */
    int frameOffset;   /* ファイルの開始フレーム */
};
```

3C

```
class Effect {
    int startTime;      /* 開始時刻(絶対フレーム) */
    int duration;       /* この効果のフレームの数 */
    int effectType;    /* 標準フォーマットの効果のタイプ(ワイプ等) */
    void *effectParams; /* 種々の効果パラメータ(スタート百分比、
                           エンド百分比、方向等) */
};
```

【図4】



【図5】

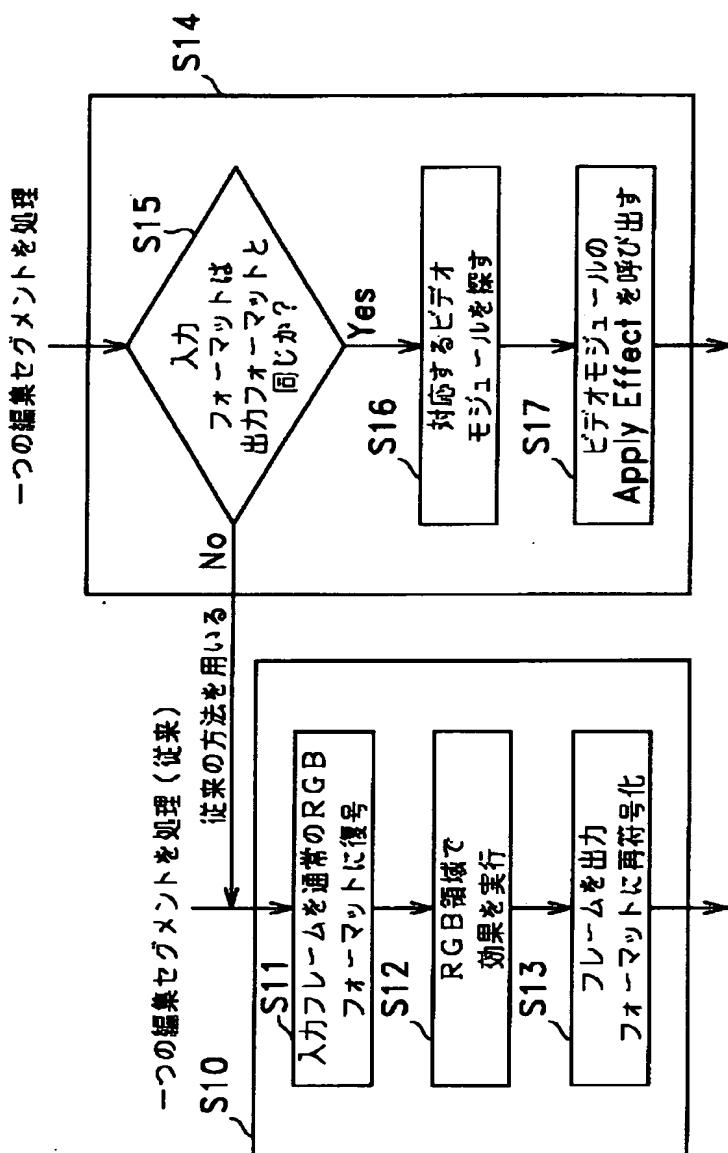
編集セグメント1	
開始	t_0
期間	$t_1 - t_0$
ビデオ1	a
ビデオ2	\emptyset
効果	b1

編集セグメント2	
開始	t_1
期間	$t_2 - t_1$
ビデオ1	a
ビデオ2	\emptyset
効果	\emptyset

編集セグメント3	
開始	t_2
期間	$t_3 - t_2$
ビデオ1	a
ビデオ2	c
効果	b2

編集セグメント4	
開始	t_3
期間	$t_4 - t_3$
ビデオ1	\emptyset
ビデオ2	c
効果	\emptyset

【図6】



【図7】

```
class VideoModel {
    int format;           /* 標準フォーマット特定子 */
    int class;            /* 標準クラス特定子 */

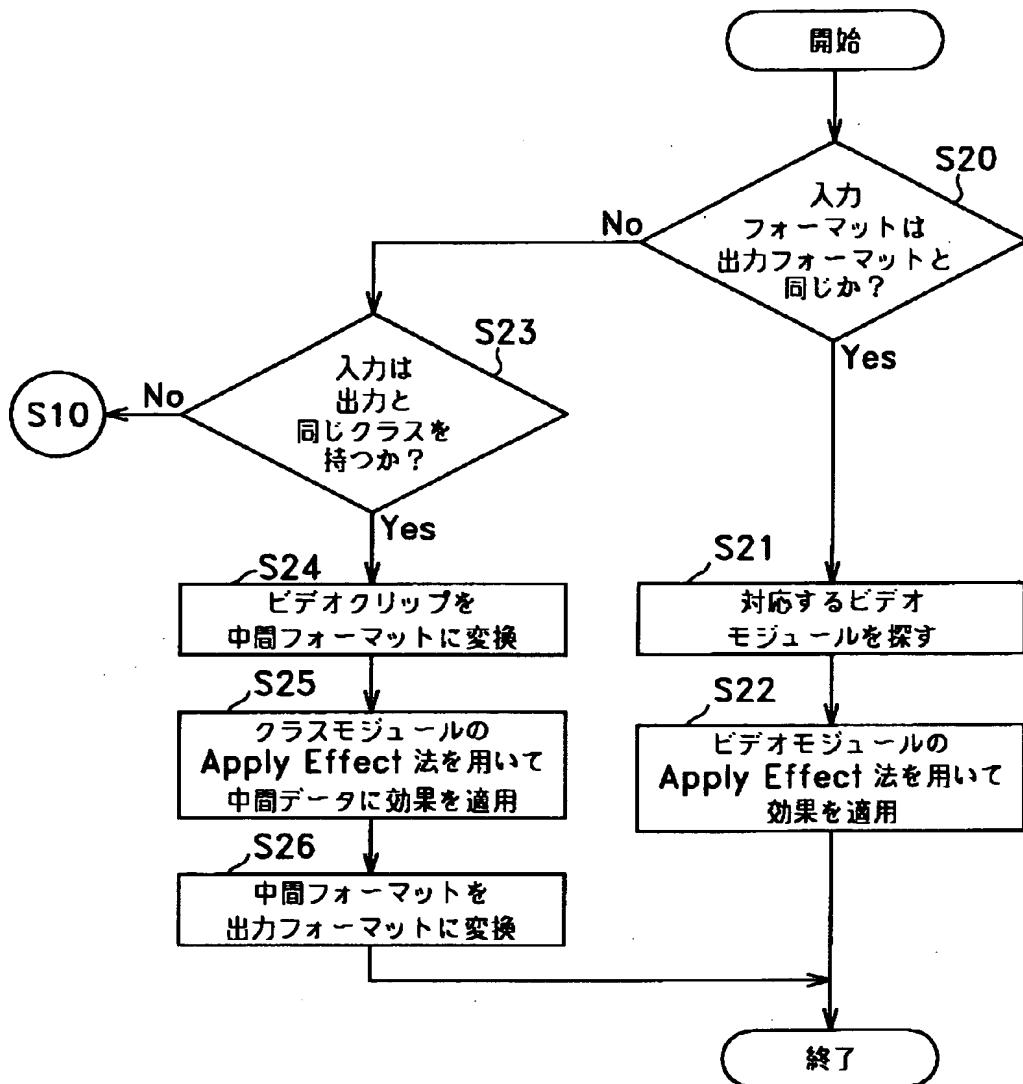
    // フレームをRGBに復号
    char *ConvertToBaseband(AVSource *pSource, int frame);

    // フレームをフォーマットに復号(例えばDCTレベルに)
    void *ConvertToClass(AVSource* pSource);

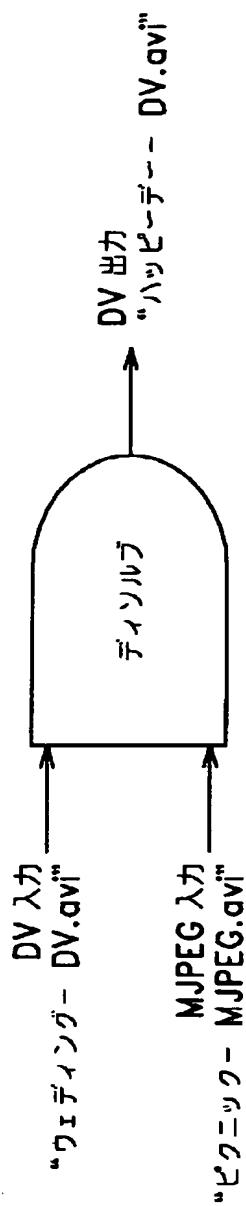
    // フォーマットのクラス(例えばDCT)からこのフォーマットに変換
    void *ConvertFromClass(Void* pClassData);

    // このフォーマットの特定の知識を用いてトランジション効果を適用
    int ApplyEffect(AVSource *pInputA, AVSource *pInputB,
                    AVSource *pOutputC, Effect *pEffect);
};
```

【図8】



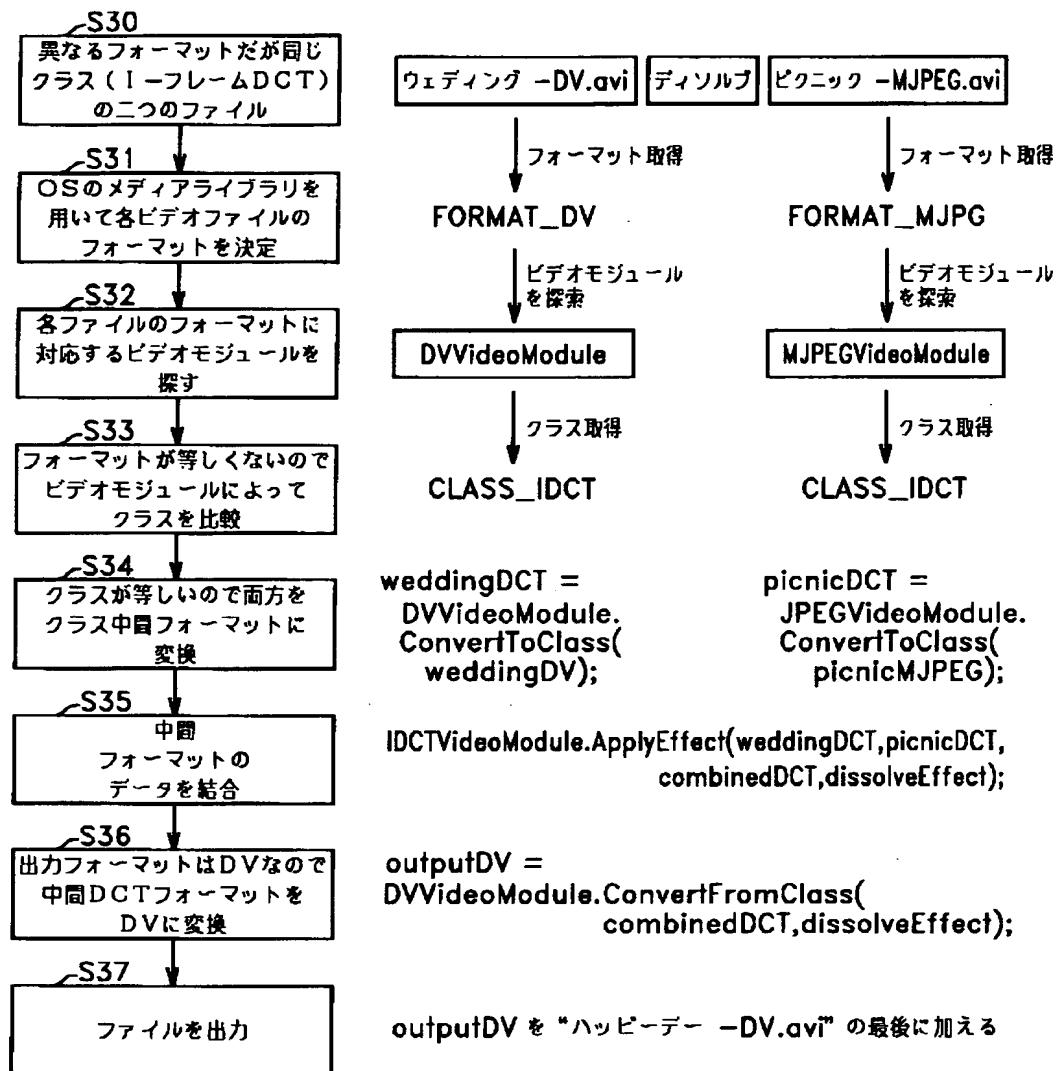
【図9】



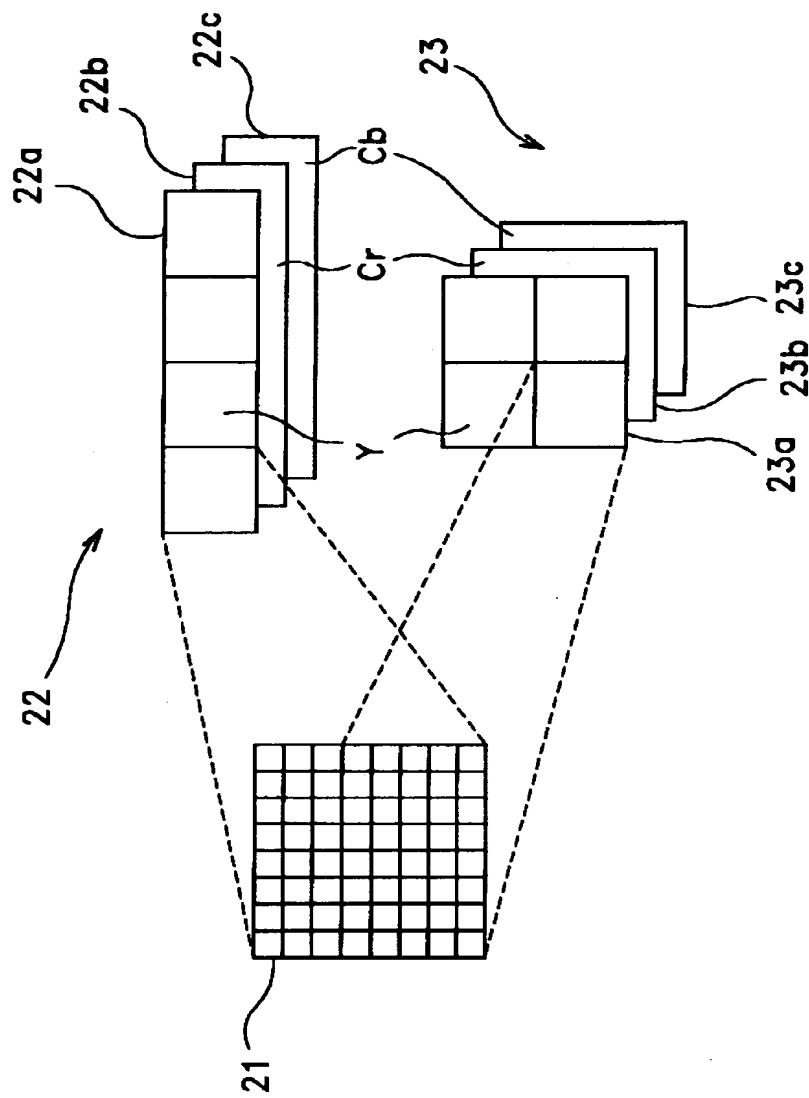
【図10】

```
#define FourCC(a,b,c,d) (((a<<24)|(b<<16)|(c<<8)|(d))  
  
#define FORMAT_DV FourCC('D','V','C','S')  
#define FORMAT_MJPG FourCC('M','J','P','G')  
  
#define CLASS_IDCT FourCC('I','D','C','T')
```

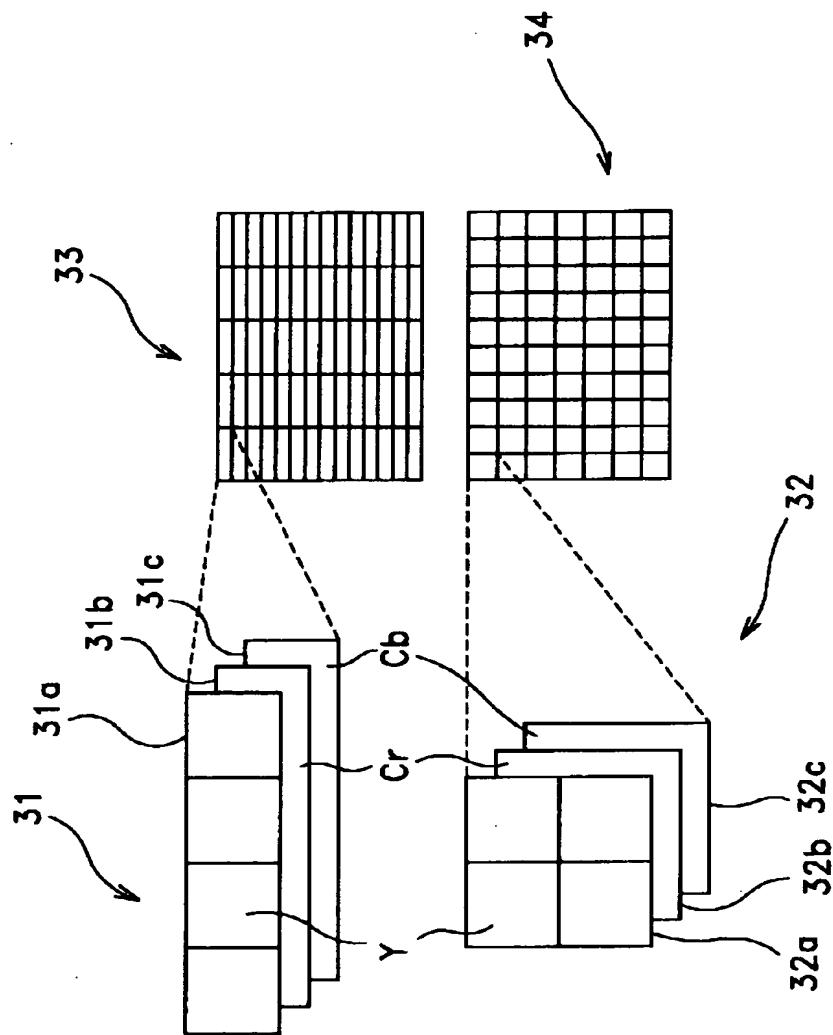
【図 1 1】



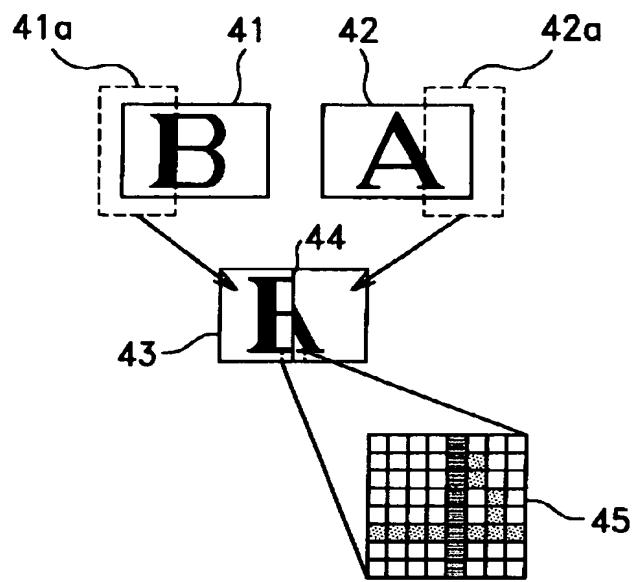
【図12】



【図13】



【図14】



【書類名】 要約書

【要約】

【課題】 画像信号の編集を高速に、画質を劣化させないように行う。

【解決手段】 状態S01においては入力コマンドが編集リストに記録され、状態S02においては編集リストが編集セグメントに分解され、状態S02においては一編集セグメントについて当該画像信号の形式に応じた処理が施され、状態S04においては一編集セグメントの処理結果が出力される。

【選択図】 図1

出願人履歴情報

識別番号 [000002185]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都品川区北品川6丁目7番35号

氏 名 ソニー株式会社